3. (Cancelled).

4. (Amended) The computer system of Claim 1, wherein the portion of the starting address includes a plurality of most significant bits of the starting address.

5. The computer system of Claim 4, wherein execution unit shifts the data elements by a predetermined number of bit positions to obtain the starting address of the cache line in which data is to be invalidated.

6. The computer system of Claim 1, wherein the predetermined portion of the plurality of cache lines is a page in the cache memory.

7. (Amended) A computer system comprising:

a first storage area to store data;

a cache memory having a plurality of cache lines each of which stores data;

a second storage area to store a data operand containing a portion of an address; and

an execution unit coupled to said first storage area, said second storage area, and said cache memory, said execution unit to operate on the portion of an address in said data operand to copy data from a predetermined portion of the plurality of cache lines in the cache memory to the first storage area, in response to receiving a single instruction of a processor instruction set.

8. The computer system of claim 7, wherein the data operand is a register location.

9. The computer system of claim 8, wherein the register location contains a plurality of most significant bits of a starting address of the cache line in which data is to be copied.

10. (Amended) The computer system of claim 9, wherein execution unit shifts the portion of an address by a predetermined number of bit positions to obtain the starting address of the cache line in which data is to be copied.

11. The computer system of Claim 7, wherein the predetermined portion of the plurality of cache lines is a page in the cache memory.

12. The computer system of Claim 7, wherein the execution unit further invalidates data in the predetermined portion of the plurality of cache lines in response to receiving the single instruction, upon copying the data to the first storage area.

13. (Cancelled).

14. (Cancelled).

15. (Cancelled).

16. (Cancelled).

17. (Cancelled).

18. (Cancelled).

19. (Cancelled).

20. (Cancelled).

21. (Cancelled).

22. (Cancelled).

23. (Cancelled).

24. (Cancelled).

25. (Cancelled).

26. (Cancelled).

27. (Cancelled).

28. (Cancelled).

29. (Cancelled).

30. (Cancelled).

31. (Cancelled).

32. (Cancelled).

33. (Cancelled).

34. (Cancelled).

35. (Cancelled).

36. (Cancelled).

37. (Cancelled).

38. A computer system comprising:

a cache memory having a plurality of cache lines each of which stores data;

a storage area to store a data operand; and

an execution unit coupled to said storage area to operate on data elements in said data operand identifying a user-definable linear or physical address identifying a predetermined portion of the plurality of cache lines to invalidate data in the predetermined portion of the plurality of cache lines in response to receiving a single cache control instruction of a processor instruction set, the single cache control instruction including a reference to the data operand.

39. The computer system of Claim 38, wherein the data operand is a register location.

40. The computer system of Claim 39, wherein execution unit shifts the data elements by a predetermined number of bit positions to obtain the starting address of the cache line in which data is to be invalidated.

41. The computer system of Claim 38, wherein the predetermined portion of the plurality of cache lines is a page in the cache memory.

42. (New) A processor comprising:

a decoder configured to decode instructions; and

a circuit coupled to said decoder, said circuit in response to a single decoded instruction of a processor instruction set being configured to:

read a portion of an address located in a register specified in the decoded instruction to obtain a starting address of a predetermined area of a cache memory on which the instruction will be performed; and

invalidate in the predetermined area of cache memory.

43. (New) The processor of Claim 42, wherein the portion of an address includes a plurality of most significant bits of the starting address.

44. (New) The processor of Claim 43, wherein the circuit shifts the portion of an address by a predetermined number of bits positions to obtain the starting address of a cache line of the predetermined area of the cache memory in which data is to be invalidated.

45. (New) The processor of Claim 42, wherein the predetermined area of the cache memory comprises a plurality of cache lines forming a page in the cache memory.

46. (New) A processor comprising:

a decoder to decode instructions, and

a circuit coupled to said decoder, said circuit in response to a single decoded instruction of a processor instruction set being configured to:

read a portion of an address located in a register specified in the decoded instruction to obtain a starting address of a predetermined area of a cache memory on which the instruction will be performed;

copy data in the predetermined area of the cache memory; and

store the copied data in storage area separate from the cache memory.

47. (New) The processor of Claim 46, wherein the portion of an address includes a plurality of most significant bits of the starting address.

48. (New) The processor of Claim 47, wherein the circuit shifts the portion of the address by a predetermined number of bit positions to obtain the starting address of a cache line of the cache memory in which data is to be copied.

49. (New) The processor of Claim 47, wherein the predetermined area comprises a plurality of cache lines forming a page in the cache memory.

50. (New) The processor of Claim 47, wherein said circuit further invalidates the data in the predetermined portion of the plurality of cache lines in response to receiving the single instruction, upon copying the data to the storage area.

51. (New) A computer-implemented method, comprising:

a) decoding a single instruction of a processor instruction set;

b) in response to said decoding of the single instruction, obtaining a portion of a starting address of a predetermined areas of a cache memory on which the single instruction will be performed by reading a portion of an address contained in a storage location specified in the decoded instruction; and

c) completing execution of said single instruction by invalidating data in the predetermined area of the cache memory.

52. (New) The method of Claim 51, wherein c) comprises setting an invalid bit corresponding to the predetermined area of the cache memory.

53. (New) The method of Claim 51, wherein b) comprises:

shifting the portion of the starting address by a predetermined number of bit positions to obtain the starting address of a cache line of the cache memory in which data is to be invalidated.

54. (New) The method of Claim 53, wherein the portion of the starting address contains a plurality of most significant bits of the starting address, and the predetermined number of bit positions represent the number of least significant bits of the starting address.

55. (New) The method of Claim 51, wherein the predetermined area is a page in the cache memory.

56. (New) A computer-implemented method, comprising:

a) decoding a single instruction of a processor instruction set;

b) in response to said decoding the single instruction, obtaining a portion of a starting address of a predetermined area of a cache memory on which the single instruction will be performed by reading a portion of an address contained in a storage location specified in the decoded instruction; and

c) completing execution of said single instruction by copying data in the predetermined area of cache memory and storing the copied data in a storage area separate from the cache memory.

57. (New) The method of Claim 56, wherein c) comprises setting an invalid bit corresponding to the predetermined area of the cache memory.

58. (New) The method of Claim 56, wherein b) comprises:

shifting the portion of the starting address by a predetermined number of bit positions to obtain the starting address of a cache line associated with the predetermined area.

59. (New) The method of Claim 58, wherein the portion of the starting address contains a plurality of most significant bits of the starting address, and the predetermined number of bit positions represent the number of least significant bits of the starting address.

60. (New) The method of Claim 56, wherein the predetermined area comprises a plurality of cache lines forming a page in the cache memory.

61. (New) The method of Claim 56, further comprises:

d) invalidating the data in the predetermined area in response to receiving the single instruction, upon copying the data to the storage area.

62. (New) A computer-readable apparatus, comprising:

a computer-readable medium that stores an instruction which when executed by a processor causes said processor to:

a) decode a single instruction of a processor instruction set;

b) in response to decoding the single instruction, obtain a portion of a starting address of a predetermined area of a cache memory on which the single instruction will be performed by reading a portion of an address contained in a storage location specified in the decoded instruction; and

c) complete execution of said single instruction by invalidating data in the predetermined area of the cache memory.

63. (New) A computer-readable apparatus comprising:

a computer-readable medium that stores an instruction which when executed by a processor causes said processor to:

a) decode a single instruction of a processor instruction set;

b) in response to decoding the single instruction, obtain a portion of a starting address of a predetermined area of a cache memory on which the single instruction will be performed by reading a portion of an address contained in a storage location specified in the decoded single instruction; and

c) complete execution of said single instruction by copying data in the predetermined area of the cache memory and storing the copied data in a storage area separate from the cache memory.

64. (New) The apparatus of Claim 63, wherein the instruction further causes the processor to:

invalidate the data in a predetermined portion of a plurality of cache lines forming the predetermined area of the cache memory in response to receiving the instruction, upon copying the data to the storage area.